

Indiana Office of Technology

MS-Access Development and Deployment Guidelines

Reference: Indiana Office of Technology (IOT)
Information Technology Standard (ITS) 99-2

Technology Profile: Software

Specific Area: Database

Document Type: Guideline

Date: October 25, 2001

Purpose: This document is intended to assist agencies in determining when development and deployment of an MS-Access multi-user database or application is appropriate.

Background: MS-Access is the state desktop database standard. MS-Access applications are typically developed and deployed as a combined application and database in its native form as .MDB files. These applications are usually developed by empowered end-users and are scaled from a single user to a small workgroup of less than 10 users. MS-Access applications may also be deployed as Microsoft Database Executable (MDE) files. MDE applications are a good choice for low-cost applications targeted for a small workgroup and may be written internally by skilled end-users and information technology professionals or externally developed.

This document provides guidelines and best practices for appropriately developing and deploying MS-Access applications. In addition, this document provides guidelines for those situations where existing MS-Access database records grow beyond its capability, or the number of users increases necessitating migration to SQL Server or Oracle.

What kind of scalability can we expect when deploying MS-Access applications and databases, and is MS-Access suitable as a database deployed for a web-based application?

When MS-Access database applications are deployed for concurrent user access, a variety of problems may result due to the following issues:

Non-RDBMS: While MS-Access is a relational database application; it lacks a defining feature of a full-scale relational database management system (RDBMS) as it does not isolate application logic from the physical data organization. MS-Access does organize data in tables and provides methods to query the data. However, this product does not separate the application logic from the physical data organization unless the application is designed appropriately in advance. Without this separation there is a dependency between the application code and the physical data organization. Data independence is a key ingredient without which application maintenance and extension are compromised. Consequently, MS-Access is not considered a full-scale RDBMS and is not suitable for mission critical or high-transaction volume applications, particularly when developed in the .mdb format.

Lack of Data Integrity and Security: Any application user who authenticates to the file-server can deliberately or inadvertently corrupt the data. In addition, write privileges to the database tables are not enforced via a user interface, and direct manipulation of the database tables is not controlled. Consequently, integrity constraints can be easily violated. If data integrity and security are important to the application, then the application must be created as an executable and security should be controlled through NTFS permissions and the built in lock-down ability of the database itself. When an MS-Access application front-ends a SQL Server or Oracle database, the security embedded within those databases must be employed.

Data Loss or No Log Recovery: MS-Access does not log transactions like other more robust databases, so data can only be recovered to the last backup. This means that any data entered in

Indiana Office of Technology

MS-Access Development and Deployment Guidelines

the database after a backup has been taken will be lost if the database becomes corrupt or the hard drive is lost. When using MS-Access, all users must be known, be notified of data loss, and be willing to accept the reentry of information. This is impractical for web-based applications since all the users cannot be identified and notified that they must reenter their data. If transaction logs and roll-back capabilities are necessary, then the backend database must be created in Oracle or SQL Server.

The Network Avalanche Effect: In MS-Access data stores, query processing takes place entirely on the desktop. This means that entire data tables must be brought to the desktop each time a query is launched. This may happen multiple times in a single query. When this type of processing involves sorting or joining larger datasets, this can cripple the network. Monitoring these transactions also becomes challenging. To reduce the network avalanche effect, the database designer should separate the application and database into linked front-end and back-end files to avoid passing the forms, etc., across the network. Queries should be carefully designed to minimize the amount of data transferred across the network and maximize the amount of data processed on the server.

Database Bloat: In general, MS-Access databases grow in size as users work on the design of their objects. Modifications to the design of a table, form, or report will cause a database to grow; compacting the database will restore it to a smaller size. However, the growth is faster and more dramatic when you change the SQL property of a query, or if you change the SQL statement of a form or report. Each time you modify a SQL statement the MS-Access database engine generates a new query plan and the new plan is saved to an unused portion of your hard disk. Database applications that frequently modify the SQL properties of objects may generate enough file growth to destabilize the application. This problem is less likely to occur in databases that use standard queries, forms and reports. Users should compact all MS-Access databases frequently, and a daily, automated database compact must be implemented for all MS-Access applications where custom query development and execution is occurring in code. An alternative strategy for applications using separate front-end and back end files is to automatically refresh the front-end application file each day from a backup.

Suitable conditions for the deployment of a MS-Access Database .mdb or MDE format using the MS-Access database (all conditions must be affirmative):

Single-file databases (.mdb or .mde format)

1. Non-mission critical applications where security is a non-issue.
2. Departments and or small work groups where data loss or corruption can be handled with minimal consequence.
3. Small databases that can be shipped across the network in their entirety with tolerable delay.
4. Small user populations (less than 10) co-located with the file server (**no WAN**).
5. Appropriate for lightly used and well-segmented LANS with a great deal of spare bandwidth.
6. Light update and query loads (a few transactions per minute only).
7. No usage for the Web.

Linked-file databases (.mdb or .mde format)

1. Non-mission critical applications requiring minimal security
2. Departments and or small work groups where data loss or corruption can be handled with minimal consequence.

Indiana Office of Technology

MS-Access Development and Deployment Guidelines

3. Larger databases that can not be shipped across the network in their entirety, or which require security, but are not so large as to require an Oracle or SQL server database.
4. Sufficient server capacity for the Back-end database
5. Small user populations (less than 10) co-located with the file server **(no WAN)**.
6. Appropriate for lightly used and well-segmented LANs with a great deal of spare bandwidth.
7. Light update and query loads (a few transactions per minute only).
8. No usage for the web.

Best Practices for MS-Access Applications using MS-Access Databases:

1. Applications should be separated into linked front-end (application) and back-end (database) files whenever possible.
2. Front-end application files should be compiled into .MDE format.
3. Back-end databases should be placed in dedicated network shares. Link specifications between the front- and back-end databases should use UNC (universal naming convention) addresses referencing the share.
4. Back end security should be provided using NTFS permissions. Front-end security should be provided using NTFS permissions and the built in lock-down ability of the database itself.
5. Applications should be designed to minimize query processing at the desktop and avoid returning entire tables across the network whenever possible.
6. Single-file applications and the back-end files of linked applications should be backed up and compacted frequently. Front-end application files should be compacted or refreshed from a backup.
7. Compact and refresh routines should be automated whenever possible.

What conditions are appropriate for use of MS-Access as an application development language fronting Oracle or SQL Server backend databases?

MS-Access applications can be utilized to query and update an external RDBMS such as Oracle or SQL Server. However, certain conditions must be considered when determining the appropriateness of MS-Access in this capacity. MS-Access is intended to provide a simple, comprehensive tool for developing small databases. The features of MS-Access that make it useful for these small databases limit its utility as an application development language. The techniques required to address security, data integrity, and the network avalanche and application bloat effects require the developer to abandon many of the built-in features of MS-Access intended to aid application developers. For this reason we do not recommend that MS-Access be used as an application development language when developing new front-end applications for other database systems.

However, we also recognize that we operate in a dynamic environment and the conditions underlying the deployment of an existing application may change quickly and with little or no warning. For example, an application database may grow beyond original expectations or an application originally intended for a small local workgroup must be shared more widely. Perhaps an application developed by an empowered user for their personal use promises productivity improvements if it can be deployed across an entire workgroup or agency. In cases like these it may be appropriate to leverage existing investments by migrating an MS-Access database to SQL Server or Oracle while continuing to use MS-Access as the application front-end.

Indiana Office of Technology

MS-Access Development and Deployment Guidelines

Suitable conditions for the deployment of an MS-Access application using an Oracle or SQL backend database (all conditions must be affirmative):

1. Non-mission critical applications requiring security.
2. The cost to replace the application exceeds the cost of renovating it. The projected life of the application should be carefully considered as part of the cost-benefit calculation
3. Updates will only be made to tables used *exclusively* by the MS-Access application.
4. Tables that are shared with other applications may be queried by the MS-Access application, but should not be updated.
5. Application is designed or modified to minimize or eliminate query processing at the desktop.
6. Security is enforced by the back-end database management system.
7. Small user populations (less than 10) co-located with the file server (**no WAN**) are still recommended as a standard guideline. However, depending on the application, the number of users may be scaled up depending on the deployment architecture and the user concurrency rate.
8. Lightly used and well-segmented LANs with a great deal of spare bandwidth.
9. Light update and query loads (a few transactions per minute only).
10. No usage for the Web.

Best Practices for MS-Access Applications using Oracle or SQL Server Databases:

1. Front-end application files should be compiled into .MDE format.
2. Application should be modified to recognize back-end security structure.
3. Data validation should be performed by the front-end application.
4. Applications should be designed to minimize query processing at the desktop and avoid returning entire tables across the network whenever possible.
5. Databases should be backed up and compacted daily. Front-end application files should be compacted or refreshed from a backup daily.
6. Compact and refresh routines should be automated whenever possible.

What conditions are appropriate for use of MS-Access as an application development language for Web-based applications?

The State of Indiana has deployed a 3-tier web architecture. In this architecture, the presentation layer, application layer and data layers are logically separated and are often hosted on different physical machines. This architecture allows the State maximum flexibility in determining the best possible hosting arrangement for any given application. MS-Access does not lend itself to this architecture as the application and data files typically need to be co-located. If MS-Access is not co-located, then the network avalanche effect is apparent and noticeable.

Also, the web server must remotely mount the appropriate drive on another server to access the database. The remote mount dependency causes concerns because a failed status of the remote mount can cause severe performance problems (including crashing) of the web server itself.

Given these architectural and operational constraints, MS-Access is not a viable tool for the development and deployment of web-based applications. Consequently, agencies are encouraged to ensure that any MS-Access application will not be deployed via the web. If it is determined that the application could grow to a point where this is required, then the agency should plan for the appropriate migration of the application, including tools, budget, and support requirements.

Indiana Office of Technology

MS-Access Development and Deployment Guidelines

If you have an existing MS-Access application, or any other desktop developed application, where the data needs to be distributed via the web, it is possible that the information can be deployed. However, this cannot be done in a real-time situation and is certainly not ideal. If you have an existing application where the data needs to be delivered via the web, contact a member of the web architecture team from **accessIndiana**, the Department of Information Technology (DoIT), or the Information Technology Oversight Commission (ITOC) so they can assist you in developing a workable and sustainable solution.

Can I use MS-Access as a querying tool for SQL compliant databases?

In the hands of sophisticated users MS-Access can be deployed as a high yield, low cost query tool against SQL compliant databases. However, like any powerful and flexible tool, it can pose significant risks to the data if it is not carefully used. MS-Access can be used to query any SQL compliant database for which an Open Database Connectivity (ODBC) driver is available. However, configuring ODBC drivers can be tricky when one is accessing an older or uncommon DBMS. While discussing the specifics of this issue is beyond the scope of this document, here are some general guidelines to consider before using MS-Access as a querying tool against larger databases.

As always, MS-Access typically processes queries on the desktop. This means that queries accessing large amounts of data can be very slow, particularly when the specification includes wildcards, calculations, or table joins. However, under certain conditions, MS-Access will allow queries to be processed directly at the server. Formulating one of these queries requires careful attention to query syntax and structure. The entire query must be written using criteria and expressions supported by the server. Including any expressions that cannot be evaluated by the server will cause the query to be processed locally. To prevent this from happening, the user should develop simple queries to abstract basic data to local, temporary MS-Access tables whenever possible. More complicated queries can then be run against these temporary tables.

Best Practices for using MS-Access as a querying tool:

1. All ODBC definitions should be set to “Read only” to avoid accidentally corrupting the data.
2. ODBC drivers must be carefully configured for optimum performance.
3. Queries should be as simple as possible, minimizing calculations, joins, or wildcards and maximizing query processing at the server.
4. Use simple queries to abstract data to local MS-Access tables for more complicated analyses.

Indiana Office of Technology

MS-Access Development and Deployment Guidelines

Summary: Agencies are cautioned to review their requirements for a specific application carefully to ensure that their return on investment is realized when utilizing MS-Access as a database as well as an application engine. MS-Access is a cost-effective tool for the development and deployment of small workgroup applications if the target audience is less than 10 users and the deployment is on a LAN. MS-Access is not an appropriate development tool for new applications that require SQL Server or Oracle databases. However, existing MS-Access applications can continue to be viable if the database and/or number of users grow beyond MS-Access' inherent database capabilities provided proper care is exercised in the conversion to a more robust database environment. MS-Access can be used as an effective querying tool for existing databases provided proper care is exercised to ensure database integrity. MS-Access is not appropriate for developing Web applications.